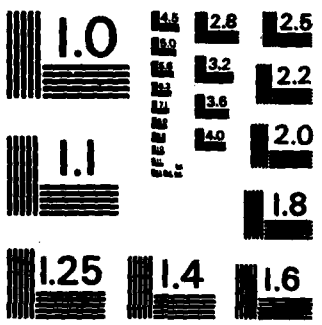
AD-A147 943 A LOCAL REFINEMENT FINITE ELEMENT METHOD FOR TIME DEPENDENT PARTIAL DIFFE..(U) RENSSELAER POLYTECHNIC INST TROY NY DEPT OF MATHEMATICAL SCIE.. 1/1

UNCLASSIFIED J E FLAHERTY ET AL. AUG 84 AFOSR-TR-84-0943 F/G 12/1 NL



REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS:
2a. SECURITY CLASSIFICATION AUTHORITY:	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE:	
4. PERFORMING ORGANIZATION REPORT NUMBER(S):	5. MONITORING ORGANIZATION REPORT NUMBER(S): AFOSR-TR- 84_0943
6a. NAME OF PERFORMING ORGANIZATION: Rensselaer Polytechnic Institute; 6b. OFFICE SYMBOL (If applicable):	7a. NAME OF MONITORING ORGANIZATION: Air Force Office of Scientific Research
6c. ADDRESS (City, State and ZIP Code): Department of Computer Science, Troy NY 12180-3590	7b. ADDRESS (City, State and ZIP Code): Directorate of Mathematical and Information Sciences, Bolling AFB DC 20332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: AFOSR; 8b. OFFICE SYMBOL (If applicable): NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: AFOSR-80-0192
8c. ADDRESS (City, State and ZIP Code): Bolling AFB DC 20332	10. SOURCE OF FUNDING NOS: PROGRAM ELEMENT NO: 61102F; PROJECT NO.: 2304; TASK NO: A3; WORK UNIT NO:

11. TITLE (Include Security Classification): A LOCAL REFINEMENT FINITE ELEMENT METHOD FOR TIME DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS

12. PERSONAL AUTHOR(S): Joseph E. Flaherty* and Peter K. Moore**

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Technical	FROM _____ TO _____	AUG 84	11

16. SUPPLEMENTARY NOTATION: *and U.S. Army Armament, Munitions & Chemical Command, Armament Research & Development Ctr, Large Calibre Weapon Systems Lab, Benet Weapons Lab, Watervliet NY 12189-5000. **Dept of Mathematical Sciences, Rensselaer Polytechnic Inst, Troy NY 12181.

17. COSATI CODES		
FIELD	GROUP	SUB GR

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Adaptive methods, finite element methods, partial differential equations; local refinement, time dependent problems.

19. ABSTRACT (Continue on reverse if necessary and identify by block number):

The authors discuss an adaptive local refinement finite element method for solving initial-boundary value problems for vector systems of partial differential equations in one space dimension and time. The method uses piecewise bilinear rectangular space-time finite elements. For each time step, grids are automatically added to regions where the local discretization error is estimated as being larger than a prescribed tolerance. The authors discuss several aspects of their algorithm, including the tree structure that is used to represent the finite element solution and grids, an error estimation technique, and initial boundary conditions at coarse-fine mesh interfaces.

The authors also present computational results for a simple linear hyperbolic problem, a problem involving Burger's equation, and a model combustion problem. Originator-supplied keywords:

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☐	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
CPT John P. Thomas, Jr.	(202) 767-5026	

A LOCAL REFINEMENT FINITE ELEMENT METHOD FOR TIME DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS [1]

Joseph E. Flaherty
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12181

and

U. S. Army Armament, Munitions, and Chemical Command
Armament Research and Development Center
Large Calibre Weapon Systems Laboratory
Benet Weapons Laboratory
Watervliet, NY 12189-5000

and

Peter K. Moore
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12181

ABSTRACT. We discuss an adaptive local refinement finite element method for solving initial-boundary value problems for vector systems of partial differential equations in one space dimension and time. The method uses piecewise bilinear rectangular space-time finite elements. For each time step, grids are automatically added to regions where the local discretization error is estimated as being larger than a prescribed tolerance. We discuss several aspects of our algorithm, including the tree structure that is used to represent the finite element solution and grids, an error estimation technique, and initial and boundary conditions at coarse-fine mesh interfaces. We also present computational results for a simple linear hyperbolic problem, a problem involving Burgers' equation, and a model combustion problem.

1. INTRODUCTION. There is an ever increasing need to solve problems of greater complexity and a corresponding need for reliable and robust software tools to accurately and efficiently describe the phenomena. Adaptive techniques are good candidates for providing the computational methods and codes necessary to solve some of these difficult problems. Two popular adaptive techniques are: (i) moving mesh methods, where a grid of a fixed number of finite difference cells or finite elements is moved so as to follow and resolve local nonuniformities in the solution, and (ii) local refinement methods, where uniform fine grids are added to coarser grids in regions where the solution is not adequately resolved. A representative sample of both types of methods is contained in Babuska, Chandra, and

---

[1] The authors were partially supported by the U. S. Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant Number AFOSR 80-0192 and the U. S. Army Research Office under Contract Number DAAG29-82-K-0197.

Flaherty [2]. Recently, Adjerid and Flaherty [1] developed a finite element method that combines mesh moving and refinement.

Herein, we discuss a local refinement finite element procedure for finding numerical solutions of M-dimensional vector systems of partial differential equations having the form

$$Lu := u_t + f(x,t,u,u_x) - [D(x,t,u)u_x]_x = 0,$$

$$a < x < b,\ t > 0, \qquad (1.1)$$

subject to the initial conditions

$$u(x,0) = u^0(x)\ ,\ a \le x \le b\ , \qquad (1.2)$$

and appropriate boundary conditions so that the problem has a well posed solution.

We discretize (1.1,2) for a time step using a finite element-Galerkin procedure with piecewise bilinear approximations on a rectangular space-time net. At the end of each time step we estimate the local discretization error, add finer subgrids of space-time elements in regions of high error, and recursively solve the problem again in these regions. The process terminates when the error estimate on each grid is less than a prescribed tolerance. The original coarse space-time grid is then carried forward for the next time step and the strategy is repeated. Our algorithm is discussed further in Flaherty and Moore [9] and some of this discussion is repeated in Section 2.

Berger [3] used a similar local refinement procedure to solve one-and two-dimensional hyperbolic systems. She used explicit finite difference schemes to discretize the partial differential equations, while we use implicit finite element techniques since we are primarily interested in parabolic problems.

In addition to the discretization technique, the major numerical questions that must be answered as part of the development of a local refinement code are (i) the estimation of the discretization error and (ii) the appropriate initial and boundary conditions to apply at coarse-fine mesh interfaces. Of course, Computer Science questions, such as which language to use to describe and implement the various algorithms and what data structures to use to represent and store the grids and solutions must also be answered. Our work in all of these areas is still far from complete and herein we only discuss our progress and thoughts on error estimation techniques, data structures, and interface conditions (cf. Section 2). In Section 3, we present the results of three examples that illustrate our method and the discussion of Section 2, and in Section 4, we present some preliminary conclusions and future plans.

2. FINITE ELEMENT ALGORITHM. We discretize equation (1.1) on a strip $\alpha < x < \beta$, $p < t < q$ using a finite element-Galerkin method with a uniform grid of N rectangular elements of size $(\beta - \alpha)/N$ by $(q - p)$. We refer to this grid as $R(\alpha,\beta,p,q,N,f,s)$, where f and s are pointers to the father and son grids discussed later. Each grid uses records to store the appropriate information.

We generate the discrete system on $R(\alpha,\beta,p,q,N,f,s)$ in the usual manner; thus, we approximate u by U(x,t) and select test functions V(x,t), where U and V are elements of a space of C^0 bilinear polynomials with respect to the grid R. We then take the inner product of equation (1.1) and V, replace u by U, and integrate any diffusive terms by parts to obtain

$$\int_R [V^T U_t + V^T f(x,t,U,U_x) + V_x^T D(x,t,U)U_x]dxdt$$

$$-\int_p^q V^T D(x,t,U)U_x|_\alpha^\beta dt = 0 . \qquad (2.2)$$

Equation (2.2) must vanish for all bilinear functions V on the grid R. The integrals are approximated using a four point Gauss quadrature rule and the resulting nonlinear system is solved by Newton iteration (cf.,e.g., [7] for additional details). Appropriate initial and boundary conditions for (2.2) are discussed later in this section.

We describe our local refinement procedure for solving problem (1.1,2) for one time step (t^0,t^1) on a coarse grid with N^0 elements, i.e, on $R(a,b,t^0,t^1,N^0,0,s)$ (where the pointer f = 0 signifies that this grid has no father). To solve this problem we simply call the procedure "locref" with the arguments $R(a,b,t^0,t^1,N^0,0,s)$, tol, tsub for each coarse grid time interval. A pseudo-PASCAL description of the procedure "locref" is shown in Figure 1.

```
procedure locref (R(α,β,p,q,N,f,s), tol, tsub)
   begin
      Solve the finite element equations (2.2) on R(α,β,p,q,N,f,s);
      Estimate the error on R(α,β,p,q,N,f,s);
      if error > tol then
         begin
            calculate where error > tol and return the son grids;
            for j := 1 to tsub do
               for i := 1 to number of sons do
                  begin
                     p[j] := p + (j-1)*(q-p)/tsub;
                     q[j] := p[j] + (q-p)/tsub;
                     locref (R(α[i],β[i],p[j],q[j],N[i],
                        R(α,β,p,q,N,f,s),s[i],tol,tsub)
                  end
         end
   end;
```

Figure 1. Algorithm for local refinement solution of (1.1,2) on $R(\alpha,\beta,p,q,N,f,s)$ with an error tolerance of tol and dividing the local time step by tsub each time the error test is not satisfied.

The recursive algorithm locref sets up a tree structure of grids with $R(a,b,t^0,t^1,N^0,0,s)$ being the root node and with the solution being

generated by a preorder traversal of the tree at each local time step. For example, if the root grid is refined to give two subgrids and the time step is halved, then the problem is solved on the first subgrid on its first time step, then on the second subgrid on the same time step, then this procedure is repeated for the second time step. The error is estimated by Richardson extrapolation, i.e., the space and time steps are halved and the problem is solved again on this new grid. The two solutions that are obtained at each original grid point are used to generate an error estimate. If this pointwise estimate exceeds the tolerance "tol", finer grids are added as leaf nodes to the tree. This procedure is similar to one used by Berger [3]; however, there are more economical error estimation strategies (cf., e.g., Bieterman and Babuska [5, 6]) which we are currently investigating.

In order to solve the finite element system (2.2) we need to supply initial and boundary conditions. On any grid with $p = 0$, $\alpha = a$, or $\beta = b$ these can be obtained from the initial condition (1.2) or prescribed boundary conditions. However, artificial initial and boundary conditions must be created at all other coarse-fine mesh interfaces. This is a difficult and crucial problem that is discussed for explicit finite difference methods by Berger [3, 4]; however, it is largely unanswered for finite element applications. Instabilities or incorrect solutions (cf. Example 1 of Section 3) can result if inappropriate conditions are specified.

For initial conditions, two strategies immediately come to mind: (i) saving all fine grid data for propagation in time or (ii) interpolating the best coarse grid data to finer grids. We consider a blend of the two strategies which consists of saving the fine grid data down to a given level λ in the tree and subsequently interpolating for finer grids. Each grid in the first λ levels either has a linked list of the initial data directly associated with it or uses an initial data list of an ancestor grid. To find the value of the solution at some new initial point, the coordinate of that point is sequentially compared to values in the linked list until an interval containing the point is found so that interpolation can be used. This is costly and we are investigating more efficient procedures that use the natural ordering that already exists. We used either piecewise linear interpolation or piecewise parabolic interpolation with shape preserving splines developed by McLaughlin [10]. For each grid in the first λ levels of the tree, a linked list is created to store the initial data. We are studying several alternative ways of determining a proper value for λ.

At the present time, we prescribe internal Dirichlet boundary conditions by linearly interpolating from coarse to finer grids. A buffer zone of two elements is added to each end of regions of high error that do not intersect the boundaries $x = a$ and b. If two buffer zones overlap or are separated from one another by one element, the two grids are joined. Similarly, if the buffer is only one element away from either a or b, that element is added to the grid.

3. NUMERICAL EXAMPLES. An experimental code based on the algorithms in Section 2 has been written in FORTRAN-77. We are testing it on several examples, some of these follow and others are presented in [9]. All results were computed in double precision on an IBM 3081D computer.

Example 1. In order to illustrate the importance of adequately resolving initial conditions at each time step we solve the linear hyperbolic initial value problem

$$u_t + u_x = 0 ,$$

$$u(x,0) = u^*(x) = \begin{cases} (1/2)(\cos(20\pi(x-0.45)) - 1) , & 0.35 < x < 0.75 \\ 0 , \text{ otherwise} \end{cases}$$

We solve this problem for one coarse time step of $\Delta t = 0.05$, 10 elements on $0 < x < 1$, tol = 0.01. For small enough times the exact solution is $u^*(x-t)$. If initial conditions are interpolated from the coarse to the fine grid, the oscillations are missed and an incorrect solution is computed, possibly without a user realizing that there is anything wrong. However, saving initial values for the first 8 levels of the tree of grids calculates the correct solution to the prescribed accuracy. The incorrect and correct solutions are shown at $t = 0.05$ in Figure 2.

Example 2. We solve the following problem for Burgers' equation:

$$u_t + uu_x = du_{xx} , \quad 0 < x < 1 , \quad 0 < t < 1 ,$$

$$u(x,0) = \sin\pi x , \quad 0 < x < 1 ,$$

$$u(0,t) = u(1,t) = 0 , \quad t > 0 .$$

We choose $d = 0.00003$, a coarse grid of 10 elements and $\Delta t = 0.1$, and piecewise parabolic approximations for the initial conditions with $\lambda = 6$. It is well known, that the solution of this problem is a "pulse" that steepens as it travels to the right until it forms a shock layer at $x = 1$. After a time of $O(1/d)$ the pulse dissipates and the solution decays to zero. We solve this problem for tol = 0.01 and 0.001 and show the solutions at $t = 0.4$ in Figure 3. The solution with the cruder tolerance is exhibiting some oscillations that are within our bounds. These, however, are not visible when the finer tolerance is used to solve the problem.

Example 3. We solve the model combustion problem

$$u_t + u_x - 2e^u = u_{xx} , \quad 0 < x < 1 , \quad 0 < t < 1 ,$$

$$u(x,0) = 0 , \quad u(0,t)=0 , \quad u_x(1,t) = 0 .$$

The exponential nonlinearity is typical in combustion problems having Arrhenius chemical kinetics. However, in this case the solution develops a "hot spot" at $x = 1$ and becomes infinite when t is approximately 0.85. We choose a coarse grid of 20 elements and $\Delta t = 0.05$, tol = 0.001, and $\lambda = 6$. In Figure 4 we show the computed solution $U(x,t)$ as a function of x for $t = 0.05$, 0.6, and 0.8 and in Figure 5 we show the mesh that was used to solve the problem. We see that the mesh is initially concentrated in the region near $x = 0$ where the curvature of the solution is largest. As time progresses and the curvature diminishes, excessive refinement is not necessary. Finally, as the solution begins to "blow-up" our algorithm generates a fine mesh only in the region near $x = 1$.

4. DISCUSSION AND CONCLUSIONS. We have briefly described an adaptive local refinement algorithm for solving time dependent partial differential equations. Even though this is very much a working algorithm, and not a production code, we are very encouraged by the preliminary results. We are investigating several possible ways of improving the efficiency and robustness of our algorithm. These include adding higher order polynomial finite element approximations, adaptively changing the number of elements that are carried forward in the coarse grid at each coarse time step, how to select the appropriate buffer length, adaptively determining the optimal number of levels of initital conditions to keep at coarse-fine interfaces, and the best boundary conditions to apply at internal boundaries. We are encouraged by the performance of McLaughlin's [10] shape preserving parabolic splines; however, the entire area of interpolating from coarse to fine grids needs further study. We are also developing non-Dirichlet "natural" boundary conditions to use at coarse-fine mesh interfaces.

Finally, we are very interested in combining the moving mesh strategy of, e.g., [7, 8] with the present local refinement strategy and extending our methods to two and three dimensions.

References

1. ADJERID, S. AND FLAHERTY, J. E., A Moving Finite Element Method for Time Dependent Partial Differential Equations with Error Estimation and Refinement, in preparation, 1984.

2. BABUSKA, I., CHANDRA, J., AND FLAHERTY, J. E. (Eds.), *Adaptive Computational Methods for Partial Differential Equations*, SIAM, Philadelphia, 1983.

3. BERGER, M. J., Adaptive mesh refinement for hyperbolic partial differential equations, Report No. STAN-CS-82-924, Department of Computer Science, Stanford University, 1982.

4. BERGER, M. J., Stability of interfaces with mesh refinement, Report No. 83-42, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, 1983.

5. BIETERMAN, M. AND BABUSKA, I., The finite elememt method for parabolic equations, I. a posteriori error estimation, *Numer. Math.*, Vol. 40, pp 339-371, 1982.

6. BIETERMAN, M. AND BABUSKA, I., The finite elememt method for parabolic equations, II. a posteriori error estimation and adaptive approach, *Numer. Math.*, Vol. 40, pp 373-406, 1982.

7. DAVIS, S. F. AND FLAHERTY, J. E., An adaptive finite element method for initial-boundary value problems for partial differential equations, *SIAM J. Sci. Stat. Comput.*, Vol. 3, pp. 6-27, 1982

8. FLAHERTY, J. E., COYLE, J. M., LUDWIG, R., AND DAVIS, S. F., Adaptive finite element methods for parabolic partial differential equations, *Adaptive Computational Methods for Partial Differential Equations*, Babuska, I., Chandra, J., and Flaherty, J. E. (Eds.), SIAM,

Philadelphia, 1983.

9. FLAHERTY, J. E. AND MOORE, P. K., An adaptive local refinement finite element method for parabolic partial differential equations, *Proc. of the International Conference on Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, Technical University of Lisbon, Lisbon, Vol. 2, pp. 139-152, 1984.

10. McLAUGHLIN, H. W., Shape preserving planar interpolation: an algorithm, *IEEE Computer Graphics and Applics.*, Vol. 3, pp 58-67, 1983.

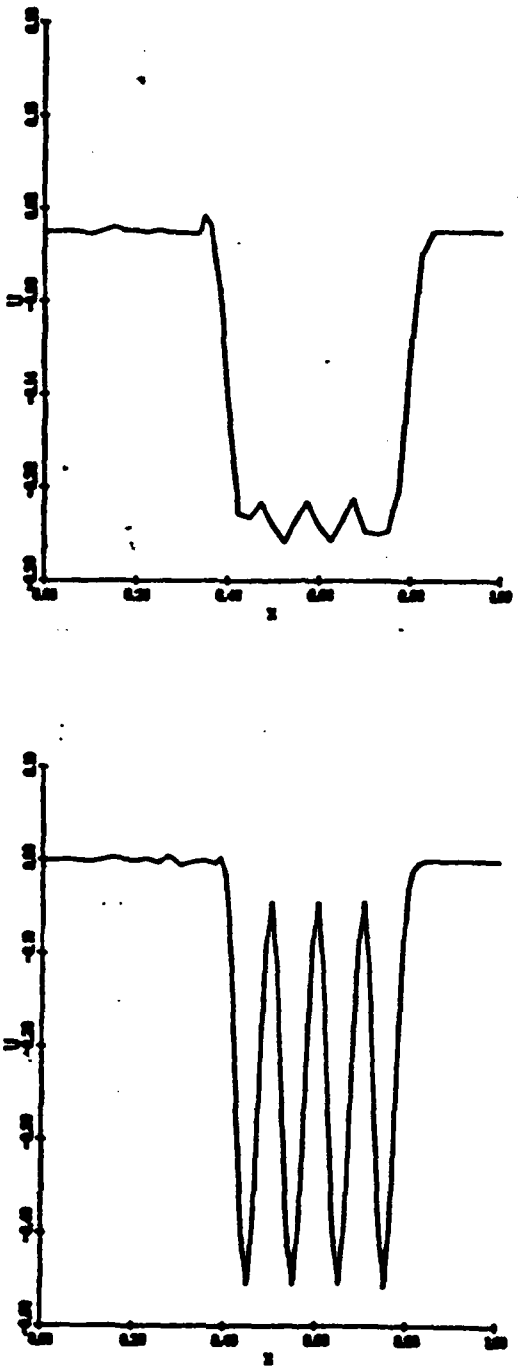


Figure 2. Solution of Example 1 at time t = 0.05 using interpolation from the coarse grid to the fine grid (top) and saving the initial values for the first 8 levels of the tree (bottom). The upper solution overlooks the oscillations and is incorrect.

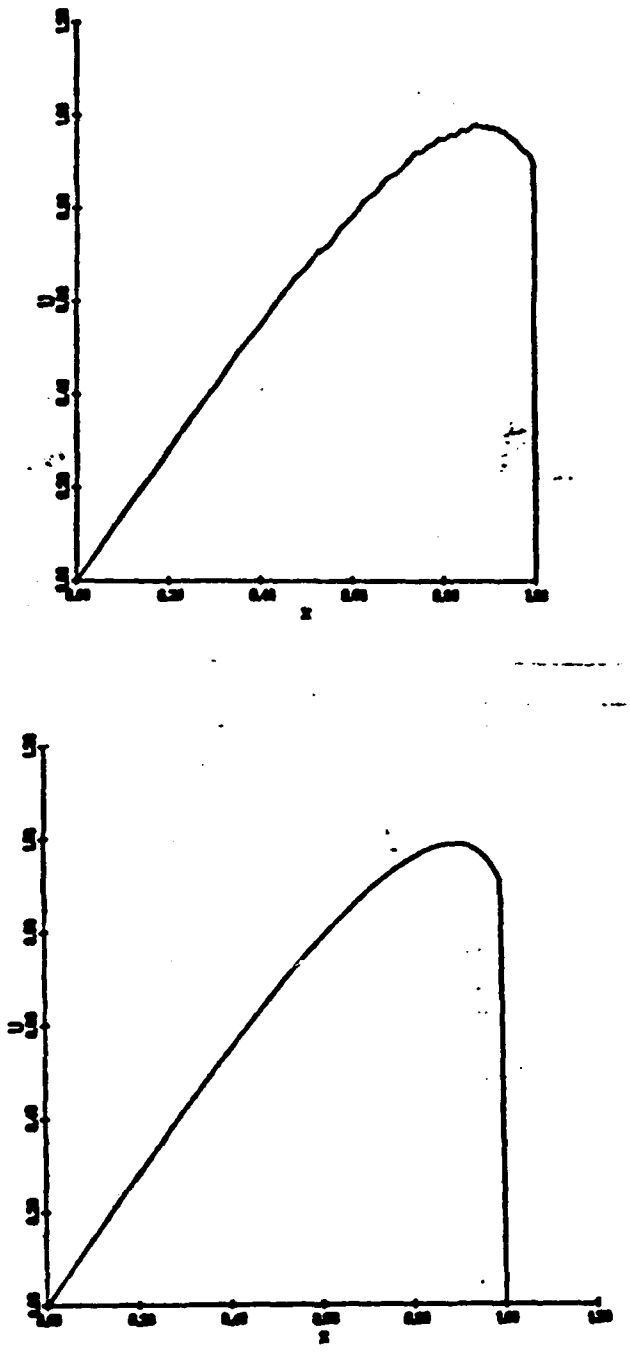


Figure 3. Solution of Example 2 at time t = 0.4 with tolerances of 0.01 (top) and 0.001 (bottom).

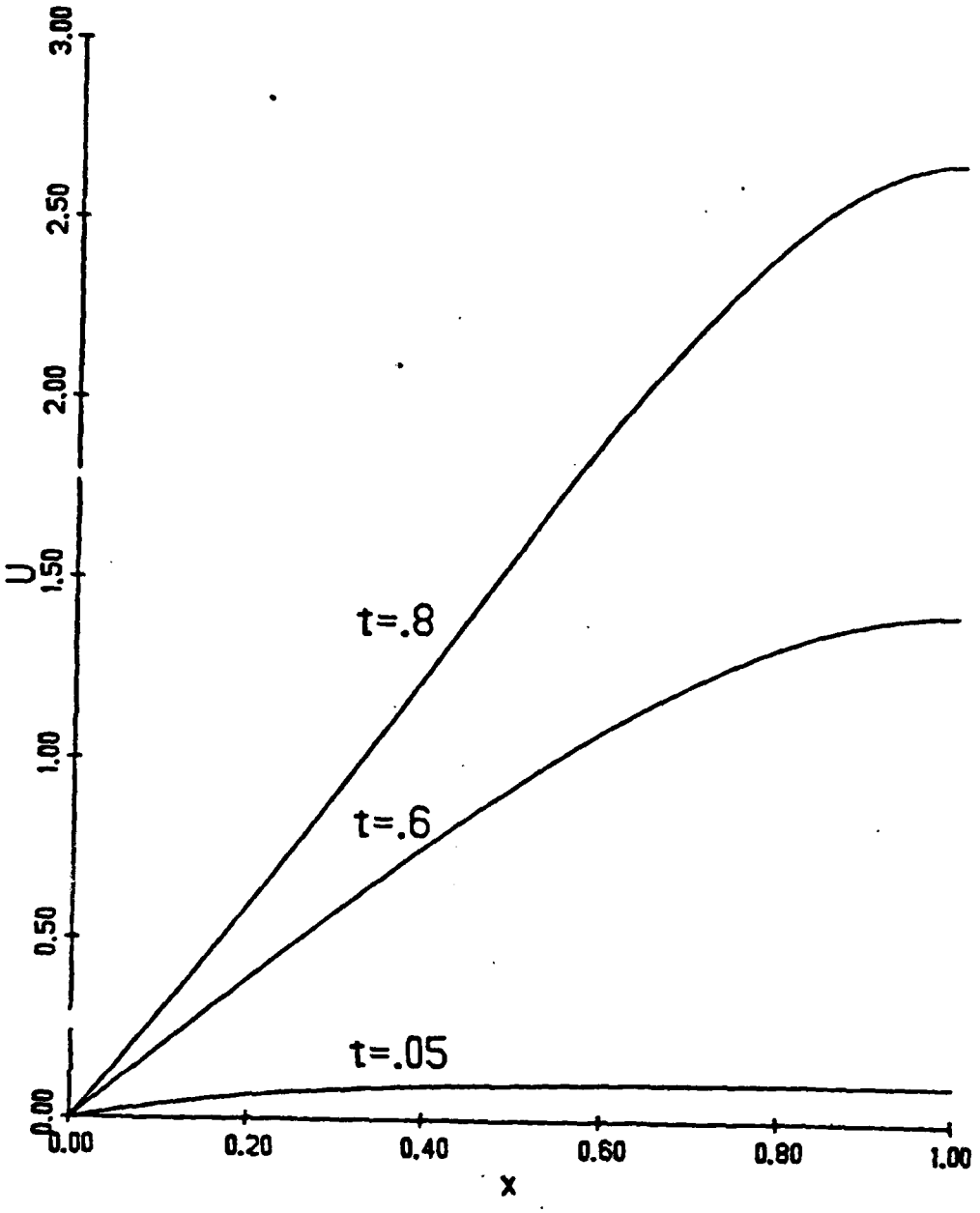
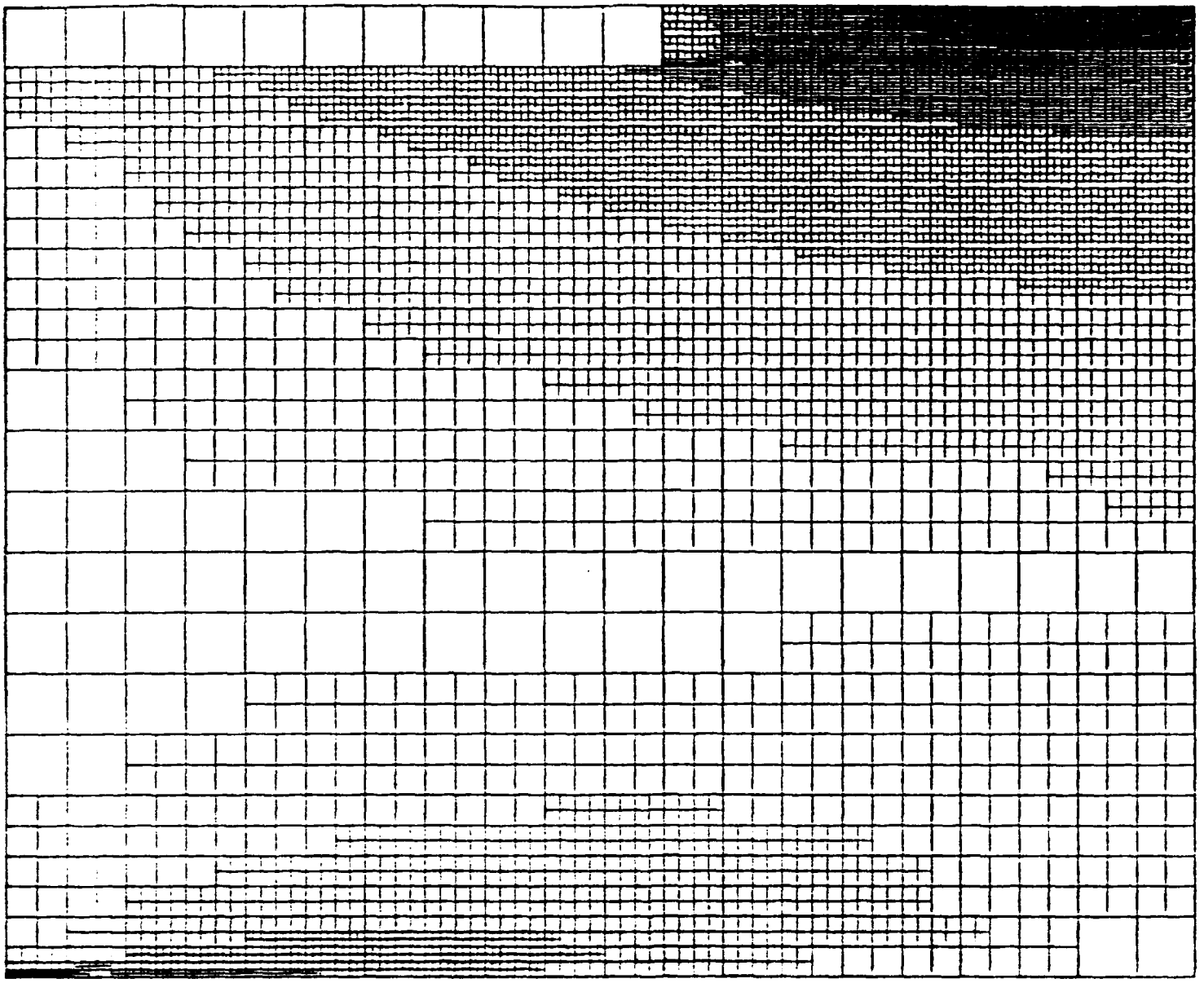
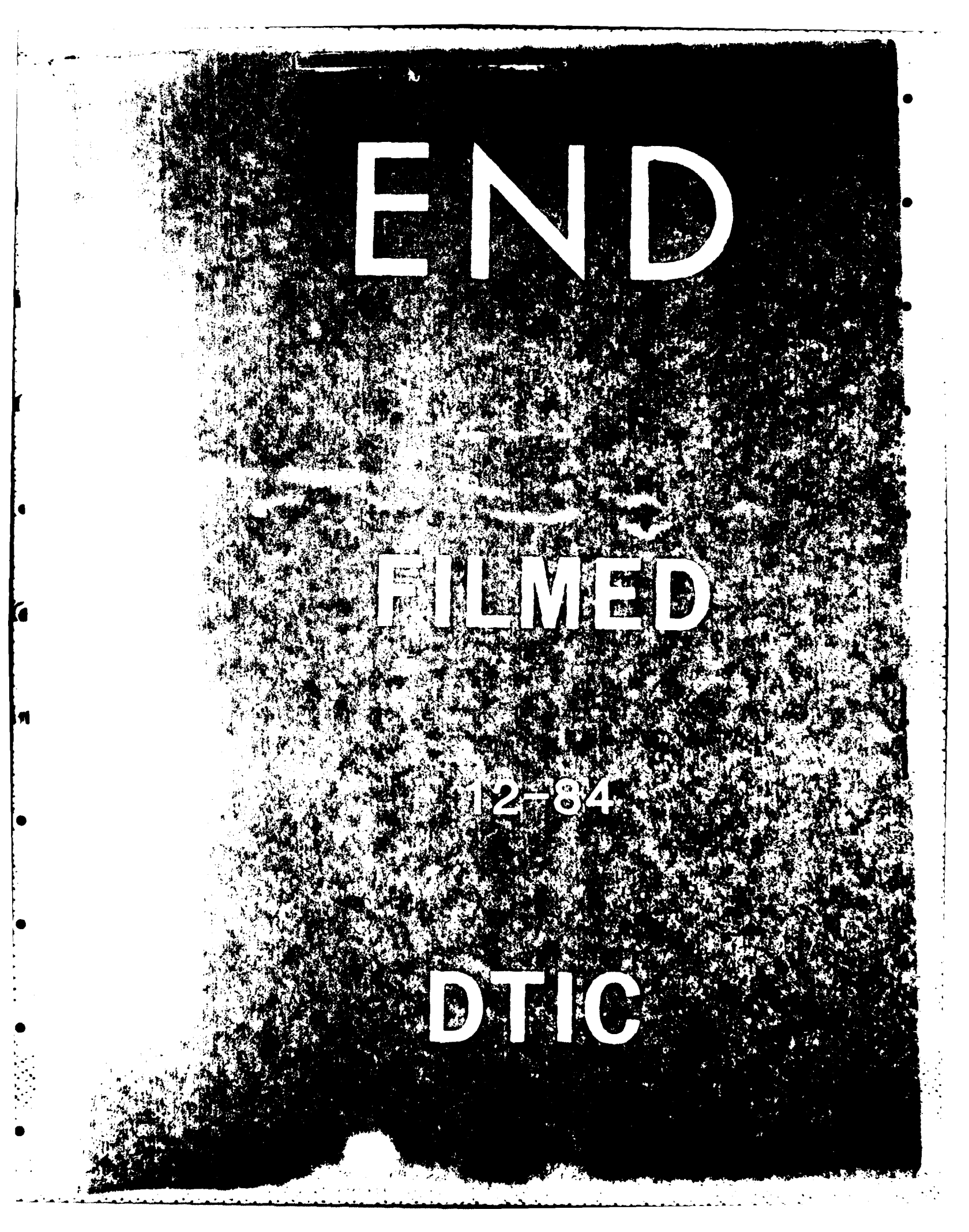


Figure 4. Solution of Example 3 at times t = 0.05, 0.6, and 0.8 with a tolerance of 0.001.



X

Figure 5. The grids generated in solving Example 3 for $0 < t < 0.8$. The initial coarse mesh has 20 elements with $\Delta t = 0.05$.


END
FILMED
12-84
DTIC